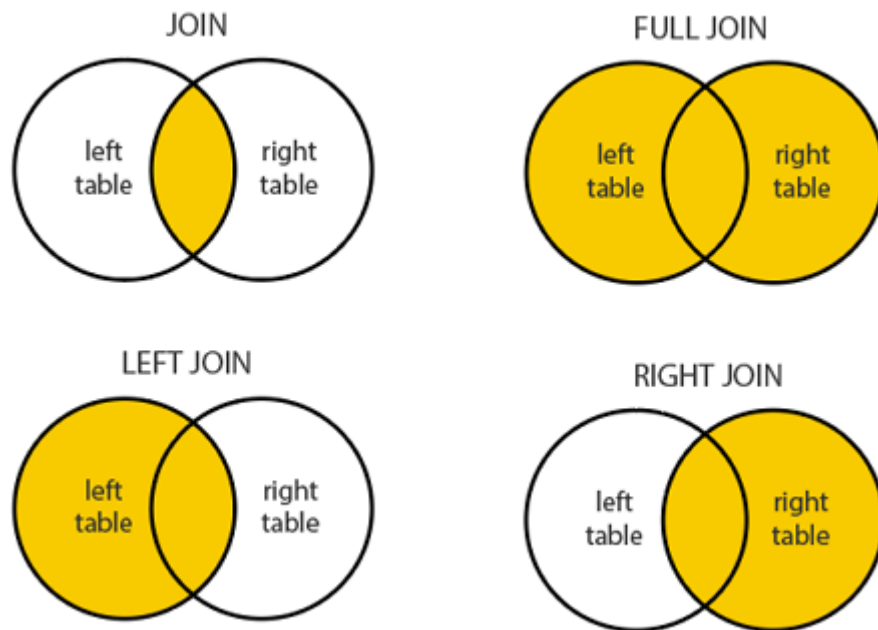


# SQL Joins

A Join in SQL is a way to get results from multiple tables which are related in some fashion. This is different from a foreign key relation. Joins are about combining data in results; foreign key relations are about enforcing referential integrity. You don't need a foreign key relation between two tables to join them for querying. A basic join looks like this:

```
SELECT *  
FROM employees  
JOIN customers ON customers.name = employees.name;
```

This will take the two tables, tableA and tableB and query them both, returning a result set that has data from both tables. The results returned will only include those rows where the value in column1 are the same in both tables. We didn't specify, so we get an INNER JOIN which is the default type of JOIN. There are others, like OUTER, LEFT, and RIGHT joins. See the image below:



The first diagram in the top left of the image is showing the type of join we saw above. To be precise, this is an INNER JOIN. The next diagram (top right) shows an OUTER JOIN. Here it is called a FULL JOIN, you may also see this called FULL OUTER JOIN. These things are all the same. Then we have LEFT and RIGHT join. Take a look at the diagrams, as they make it fairly clear what data is returned in our result set.

Whenever we query the database, we always query at least one table. This would be considered the "left table" in these diagrams. The table we are adding in would be the "right table". LEFT and RIGHT are keywords in SQL for performing joins, and refer to the initial table, and the added table. We can add multiple tables too, with multiple join clauses. If we start with tableA, join in tableB, we could then join in tableC treating tableB as the "left" table. It's confusing, just know that "left" and "right" are just referring to which table is which. There is no physical left or right here.

# SQL Joins

Let's consider some examples to drive the point home. Consider the following tables:

**customers table:**

	name	address
1	Terry Crews	1 Way st. Beverly Hills, CA 90210
2	Andy Samberg	21 Jump st. Chicago, IL 60652
3	Melissa Fumero	123 Avenue B New York, NY 10451
4	Joe Lo Truglio	226 Andy Street Rapid Valley, SD 57701

**employees table:**

1	Terry Crews	1 Way st. Beverly Hills, CA 90210
2	Andy Samberg	21 Jump st. Chicago, IL 60652
3	Andre Braugher	1031 Stanley Avenue Hempstead, NY 11550
4	Stephanie Beatriz	1004 Pin Oak Drive Whittier, CA 90606

Note that both tables include Terry Crews and Andy Samberg. Note that each table also includes two names which do not appear in the other table. Now let's perform several different join operations on these two tables and see the results.

```
--Inner join
--get only the data from both tables where they match
--this is the default join
SELECT *
FROM employees
INNER JOIN customers ON customers.name = employees.name;
```

	name	address		name	address
1	Terry Crews	1 Way st. Beverly Hills, CA 90210	1	Terry Crews	1 Way st. Beverly Hills, CA 90210
2	Andy Samberg	21 Jump st. Chicago, IL 60652	2	Andy Samberg	21 Jump st. Chicago, IL 60652

```
--Left join
--get data from left (original) table
--include matching data from right (joined) table
SELECT *
FROM employees
LEFT JOIN customers ON employees.name = customers.name;
```

	name	address		name	address
1	Terry Crews	1 Way st. Beverly Hills, CA 90210	1	Terry Crews	1 Way st. Beverly Hills, CA 90210
2	Andy Samberg	21 Jump st. Chicago, IL 60652	2	Andy Samberg	21 Jump st. Chicago, IL 60652
3	Andre Braugher	1031 Stanley Avenue Hempstead, NY 11550		[NULL]	[NULL]
4	Stephanie Beatriz	1004 Pin Oak Drive Whittier, CA 90606		[NULL]	[NULL]

# SQL Joins

```
--Right join
--get data from the right (joined) table
--include matching data from the left (original) table
SELECT *
FROM employees
RIGHT JOIN customers ON employees.name = customers.name;
```

12	ABC	name	ABC	address	12	ABC	name	ABC	address
1		Terry Crews		1 Way st. Beverly Hills, CA 90210	1		Terry Crews		1 Way st. Beverly Hills, CA 90210
2		Andy Samberg		21 Jump st. Chicago, IL 60652	2		Andy Samberg		21 Jump st. Chicago, IL 60652
		[NULL]		[NULL]	3		Melissa Fumero		123 Avenue B New York, NY 1045
		[NULL]		[NULL]	4		Joe Lo Truglio		226 Andy Street Rapid Valley, SD

```
--Outer join / Full join / Full Outer Join - these are all the same thing
--get all data from both tables regardless of matching
--any non-matching rows will have null values
SELECT *
FROM employees
FULL JOIN customers ON employees.name = customers.name;
```

12	ABC	name	ABC	address	12	ABC	name	ABC	address
1		Terry Crews		1 Way st. Beverly Hills, CA 90210	1		Terry Crews		1 Way st. Beverly Hills, CA 90210
2		Andy Samberg		21 Jump st. Chicago, IL 60652	2		Andy Samberg		21 Jump st. Chicago, IL 60652
3		Andre Braugher		1031 Stanley Avenue Hempstead, NY 11540			[NULL]		[NULL]
4		Stephanie Beatriz		1004 Pin Oak Drive Whittier, CA 90601			[NULL]		[NULL]
		[NULL]		[NULL]	3		Melissa Fumero		123 Avenue B New York, NY 1045
		[NULL]		[NULL]	4		Joe Lo Truglio		226 Andy Street Rapid Valley, SD

# SQL Joins

Here is the script you can use to try this out on your own:

```
CREATE TABLE customers (  
    customer_id INT PRIMARY KEY,  
    name VARCHAR(200),  
    address VARCHAR(200)  
);  
  
CREATE TABLE employees (  
    employee_id INT PRIMARY KEY,  
    name VARCHAR(200),  
    address VARCHAR(200)  
);  
  
INSERT INTO customers VALUES (1, 'Terry Crews', '1 Way st. Beverly Hills, CA 90210');  
INSERT INTO customers VALUES (2, 'Andy Samberg', '21 Jump st. Chicago, IL 60652');  
INSERT INTO customers VALUES (3, 'Melissa Fumero', '123 Avenue B New York, NY 10451');  
INSERT INTO customers VALUES (4, 'Joe Lo Truglio', '226 Andy Street Rapid Valley, SD 57701');  
  
INSERT INTO employees VALUES (1, 'Terry Crews', '1 Way st. Beverly Hills, CA 90210');  
INSERT INTO employees VALUES (2, 'Andy Samberg', '21 Jump st. Chicago, IL 60652');  
INSERT INTO employees VALUES (3, 'Andre Braugher', '1031 Stanley Avenue Hempstead, NY 11550');  
INSERT INTO employees VALUES (4, 'Stephanie Beatriz', '1004 Pin Oak Drive Whittier, CA 90606');  
  
SELECT * FROM customers;  
SELECT * FROM employees;  
  
--Inner join  
--get only the data from both tables where they match  
--this is the default join  
SELECT *  
FROM employees  
INNER JOIN customers ON customers.name = employees.name;  
  
--Left join  
--get data from left (original) table  
--include matching data from right (joined) table  
SELECT *  
FROM employees  
LEFT JOIN customers ON employees.name = customers.name;  
  
--right join  
--get data from the right (joined) table  
--include matching data from the left (original) table  
SELECT *  
FROM employees  
RIGHT JOIN customers ON employees.name = customers.name;  
  
--Outer join/Full join  
--get all data from both tables regardless of matching  
--any non-matching rows will have null values  
SELECT *  
FROM employees  
FULL JOIN customers ON employees.name = customers.name;
```